

AFRL-IF-RS-TR-2003-87
Final Technical Report
April 2003



JAVA-BASED PERFORMANCE ORIENTED VISUALIZATION SYSTEM

Solipsys Corporation

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK**

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2003-87 has been reviewed and is approved for publication.

APPROVED:



WILLIS J. HORTH
Project Engineer

FOR THE DIRECTOR:



JOSEPH CAMERA, Chief
Information & Intelligence Exploitation Division
Information Directorate

REPORT DOCUMENTATION PAGE			<i>Form Approved</i> OMB No. 074-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE APRIL 2003	3. REPORT TYPE AND DATES COVERED Final Apr 02 – Jan 03	
4. TITLE AND SUBTITLE JAVA-BASED PERFORMANCE ORIENTED VISUALIZATION SYSTEM			5. FUNDING NUMBERS C - F30602-02-C-0058 PE - 65502F PR - 3005 TA - 21 WU - 07	
6. AUTHOR(S) Eric S. Conn				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Solipsys Corporation 6100 Chevy Chase Drive Laurel Maryland 20707-2929			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/IFEA 32 Brooks Road Rome New York 13441-4114			10. SPONSORING / MONITORING AGENCY REPORT NUMBER AFRL-IF-RS-TR-2003-87	
11. SUPPLEMENTARY NOTES AFRL Project Engineer: Willis J. Horth/IFEA/(315) 330-3430/ Willis.Horth@rl.af.mil				
12a. DISTRIBUTION / AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.				12b. DISTRIBUTION CODE
13. ABSTRACT (Maximum 200 Words) Report Developed Under SBIR The Solipsys Tactical Display Framework (TDF) is a Java-based, performance-oriented visualization system that meets the stringent requirements of a Dynamic Battle Management (DBM) environment for Command and Control (C2) TDF has been selected as the Graphical User Interface (GUI) visualization system for AWACS 40/45. The USAF is interested in determining whether it is applicable to other C2 platforms such as JSTARS, MCE, and new efforts such as MC2A. This report provides a summary of the program contacted and their reaction to TDF.				
14. SUBJECT TERMS SBIR Report, Java Based Visualization System, Application Programmer Interface			15. NUMBER OF PAGES 14	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	
NSN 7540-01-280-5500			Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. Z39-18 298-102	

TABLE OF CONTENTS

1	BACKGROUND	1
2	PROJECT OBJECTIVES	1
3	WORK PERFORMED	1
4	RESULTS OBTAINED.....	2
5	TECHNICAL FEASIBILITY	3
5.1	THE TDF DESIGN APPROACH.....	6
5.2	THE TDF SOFTWARE ORGANIZATION	6
5.2.1	<i>Concept of Execution</i>	7
5.2.2	<i>Model/View</i>	7
5.2.3	<i>Object/Listener</i>	7
5.2.4	<i>Factory Paradigm</i>	9
5.2.5	<i>User Features.....</i>	9
6	CONCLUSION	9

List of Figures

FIGURE 1 – BENEFITS AND FEATURES OF TDF	4
FIGURE 2 – TDF SINGLE INTEGRATED AIR PICTURE (SIAP) FOR NORAD	5
FIGURE 3 – AUTOMATIC ACO/ATO OVERLAY CREATION IN TDF	5
FIGURE 4 – TDF MODEL/VIEW DESIGN PATTERN.....	8
FIGURE 5 – TDF OBJECT/LISTENER DESIGN PATTERN	8
FIGURE 6 – TDF FACTORY DESIGN PARADIGM.....	9
FIGURE 7 – HIGH-LEVEL FEATURES OF TDF	10

List of Tables

TABLE 1 - ORGANIZATIONS CONTACTED DURING SBIR PHASE I.....	2
--	---

1 BACKGROUND

Solipsys has developed a modular, standards-based, open-architecture visualization product called the Tactical Display Framework (TDF) that has been certified by Sun Microsystems as “100% Pure Java” and meets or exceeds the usability and performance requirements of many Command and Control (C2) applications. TDF is a licensed enabling software technology that promotes third-party development and extension through a sophisticated and well-documented Application Programmer Interface (API) and has a large customer base in the Department of Defense (DOD) and internationally. During the summer of 2002, the United States Air Force (USAF) and The Boeing Company officially selected TDF, known within program circles as the Primary Airborne Warning and Command System (AWACS) Display (PAD), as the Human-Machine Interface (HMI) for AWACS 40/45. This decision was the culmination of over two years of detailed engineering and operational analysis, and included extensive input from the user community. Given that a suitable Java-based, performance-oriented visualization system existed and the fact that it had already been selected for AWACS, the (Small Business Innovative Research) SBIR Phase I program manager for this topic redirected the focus of the contract from the proposed third-party HMI development effort into a technology demonstration and feasibility study for other key USAF C2 programs such as Modular Control Equipment (MCE), Joint Surveillance Target Acquisition Radar System (JSTARS), Global Hawk, and Intelligence, Surveillance, and Reconnaissance (ISR) Manager. This final report documents the results of the program-specific demonstrations and feasibility studies and recommends a path for achieving the desired Common Battle Management Software (CBMS) vision using TDF as the HMI component.

2 PROJECT OBJECTIVES

The primary objective during Phase I of this SBIR contract was to expose key C2 platforms and programs to the TDF technology and to elicit feedback on its applicability now and in the future. Since TDF already has an established market presence, most of the people contacted were generally aware of the product's capabilities and welcomed the opportunity to learn more details. The types of personnel contacted included program managers, operators, software developers, and the test community. Program managers and operational staff were most often given a TDF presentation followed by a demonstration whereas technical personnel were provided Application Programmer Interface (API)-based development kits and sample code for their evaluation. The objective of the effort was to generate feedback on the effectiveness and usability of the TDF software for other programs.

3 WORK PERFORMED

The time and resources devoted to each C2 program differed depending on many factors including program need, scheduling constraints, and initial perceptions. **Table 1** shows the organizations that were contacted and provides a synopsis of the type of interaction. Each organization listed in the table below was very receptive to TDF and positive about its use as the common look-and-feel on its programs.

Organization/Program	Location	Type of Interaction
AFRL/HEC	Wright Patterson AFB	On-site visit, demonstration, email, phone
AFRL/HECP (Advanced UAV Interfaces)	Wright Patterson AFB	Email, third-party demonstration

Organization/Program	Location	Type of Interaction
JSTARS Joint Test Force	Northrop Grumman – Melbourne	On-site visit, demonstration, email, phone
AFC2ISRC/TC	Langley AFB	Multiple on-site visits, demonstrations, limited ISR-M MTIX integration, email, phone
ACC/DRRG	Langley AFB	On-site visit, demonstration, email, phone
TBMCS	Lockheed Martin Mission Systems	Email, phone, third-party demonstration
84 RADES/SCD	Hill AFB	On-site visit, demonstration, email, phone
USAFSAM/FEP	Brooks AFB	Email, phone

Table 1 - Organizations Contacted During SBIR Phase I

4 RESULTS OBTAINED

It became clear during the course of the Phase I contract that many in the C2 community view TDF as the best technology for achieving an intuitive common look-and-feel for both ground and airborne platforms. The benefits of a common look-and-feel include significant reductions in procurement and maintenance costs, reduced training requirements as operators move from platform to platform, and an anticipated reduction in operational errors.

Although not associated with this SBIR Phase I effort, the following USAF programs and/or communities are currently using TDF as their common look-and-feel:

- AWACS 40/45
- Nellis AFB 98th Range Wing
- Nellis AFB Weapons School
- Korean Tactical Data Information Link (TADIL) Architecture Improvement Program (KTAIP)
- North American Aerospace Defense Command (NORAD) Contingency Suite (NCS)
- Alaskan Aerospace Surveillance and Range Operations Modernization (AASROM)
- Battle Control Center – Experimental (BCC-X)
- Joint Distributed Engineering Plant (JDEP)
- Command and Control Alternate Capability (C2AC)
- JSTARS Advanced Prototyping
- Theatre Battle Management Core System (TBMCS) Advanced Prototyping

- Non-Organic Radar Access (NORA)
- Advanced Cruise Missile Defense (ACMD) Advanced Concept Technology Demonstration (ACTD)
- Distributed Mission Training (DMT)
- Network-Centric Collaborative Targeting (NCCT) ACTD
- Project Suter

There are many Joint and International customers for TDF. For instance, the Royal Australian Air Force (RAAF) is currently using TDF as the basis of their country's air defense system in a program called the Interim Tracking and Display Software Air Defense System (ITDSA). The Italian Navy also has employed TDF for surface-based surveillance and border protection.

As a result of interaction with the Air Force C2 ISR Center (AFC2ISRC) during the course of this SBIR, work has begun to integrate TDF into the ISR Manager and Moving Target Information Exploitation (MTIX) systems. The anticipated result is a single display that provides real-time information on all ground and air targets within an area of interest.

The Boeing Company and the AFRL Human Effectiveness Directorate at Wright Patterson AFB are also evaluating TDF as the surveillance and control HMI for Global Hawk. Much of this new work is a direct result of the exposure to TDF that started under Phase I of this SBIR. Finally, the JSTARS and ground communities are also beginning to explore the use of TDF as their front-end HMI and results obtained thus far have been positive.

5 TECHNICAL FEASIBILITY

Key to the success of large-scale third-party development activities like the AWACS PAD is the well-documented, object-oriented API provided as part of the TDF Developer's Kit. It provides direct access to all of the features commonly present within modern C2 and surveillance systems such as tracks, sensors, maps, tactical data links, and geographic coordinate systems. The TDF development environment extends the inherent Sun Java APIs through a dynamic plug-in architecture that enables programmers unfamiliar with the product to excel in a relatively short period of time. It is common for experienced Java programmers to become productive within days and deploy moderately complex applications within a week or two from initial exposure to the toolkit.

The API incorporates elements of the de facto standard Java Beans paradigm for component-based software and facilitates the development of dynamically loaded, customized plug-in objects that can be seamlessly distributed across a network. Many of the features available to operational users and software developers through TDF's API are provided in **Figure 1**.

The TDF poses little technical risk; it is currently being employed as the core graphics system for many ground and airborne C2 systems. It has demonstrated real-time performance under heavy loading conditions (>17,000 tracks/second, >20,000 plots/sec, 116 radar interfaces) and supports automatic incorporation of Air Control Order (ACO) and Air Tasking Order (ATO) data. **Figure 2** shows the TDF Single Integrated Air Picture (SIAP) generated as part of the NCS that is operational 24x7 at three Continental United States (CONUS) Air Defense Sectors. **Figure 3** depicts ACO/ATO overlays for Nellis AFB that were automatically created from Theater Battle Management Core System (TBMCS) United States Message Text Format (USMTF) data.

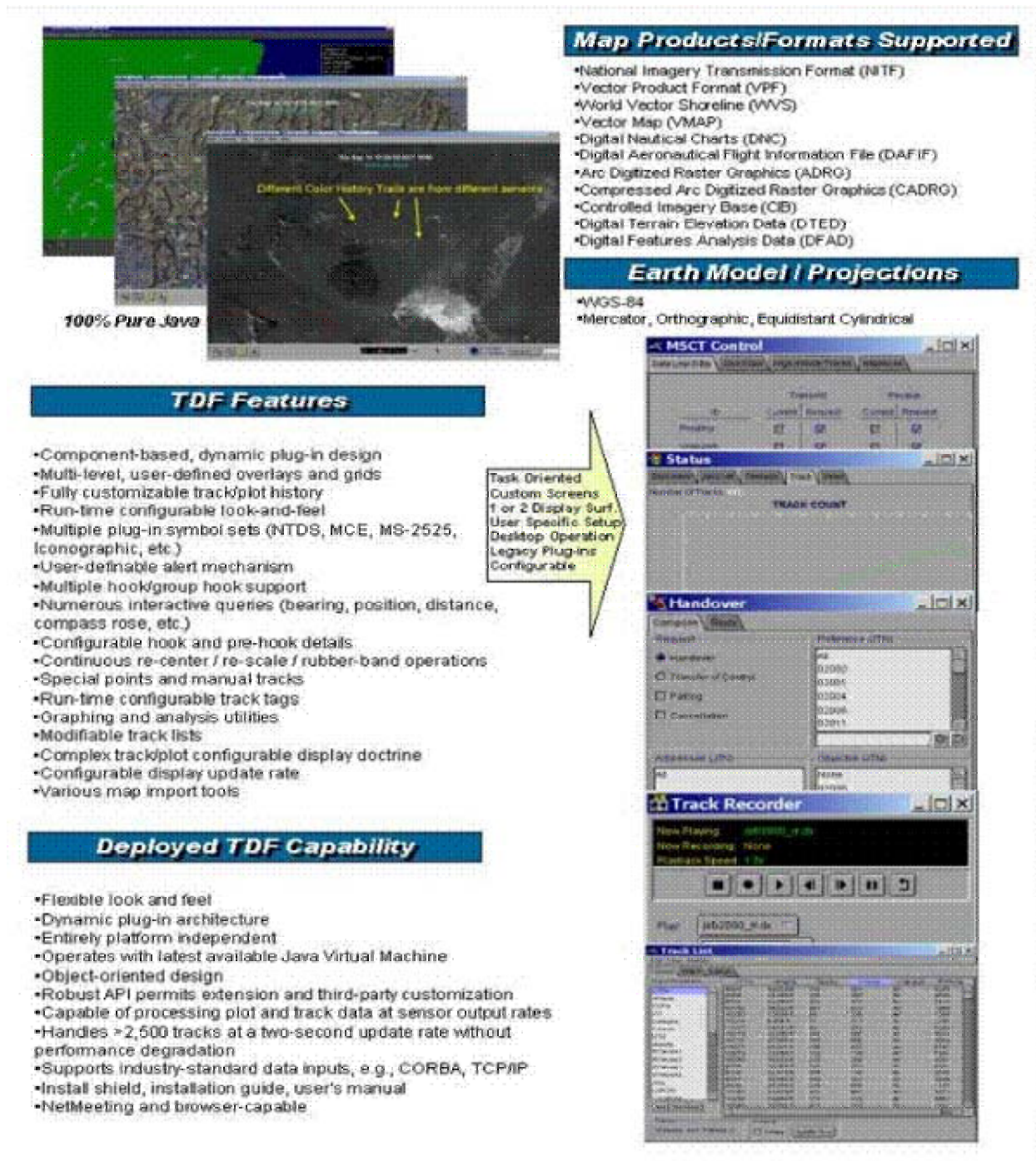


Figure 1 - Benefits and Features of TDF

Because of its modern open architecture Java-based implementation, TDF is fully compliant with all Government software architectures including the Defense Information Infrastructure Common Operating Environment (DII COE), the C2 Enterprise Reference Architecture (C2ERA), and the Joint Technical Architecture (JTA).

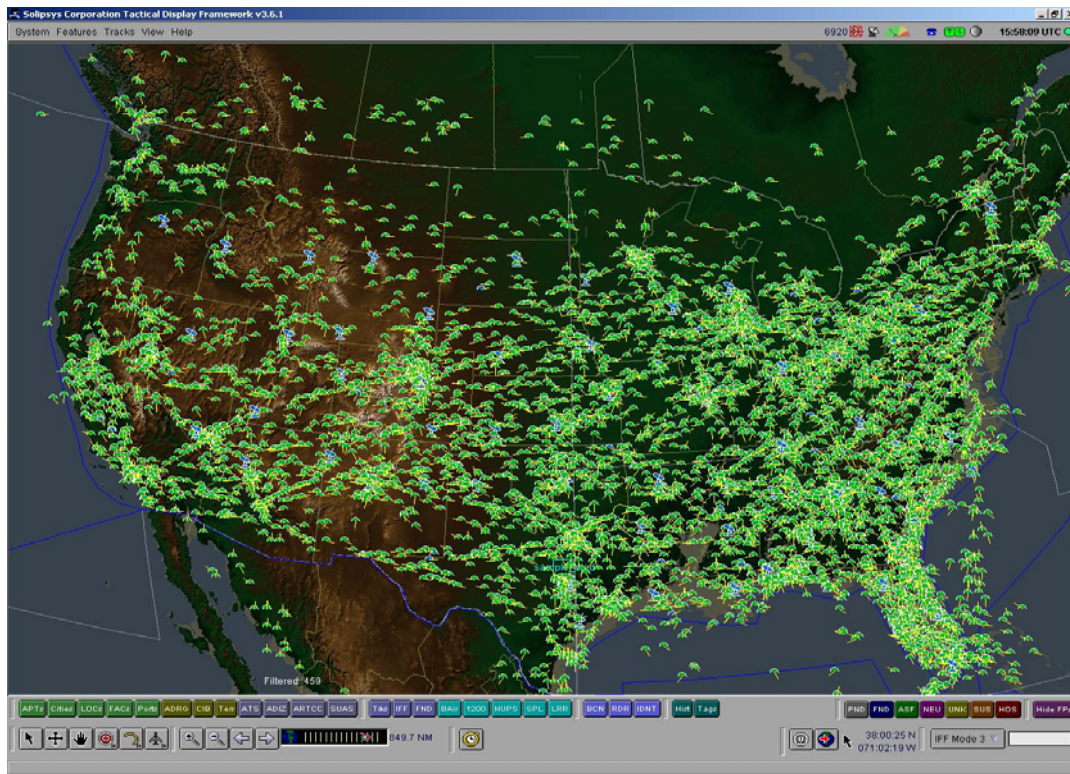


Figure 2 - TDF Single Integrated Air Picture (SIAP) for NORAD

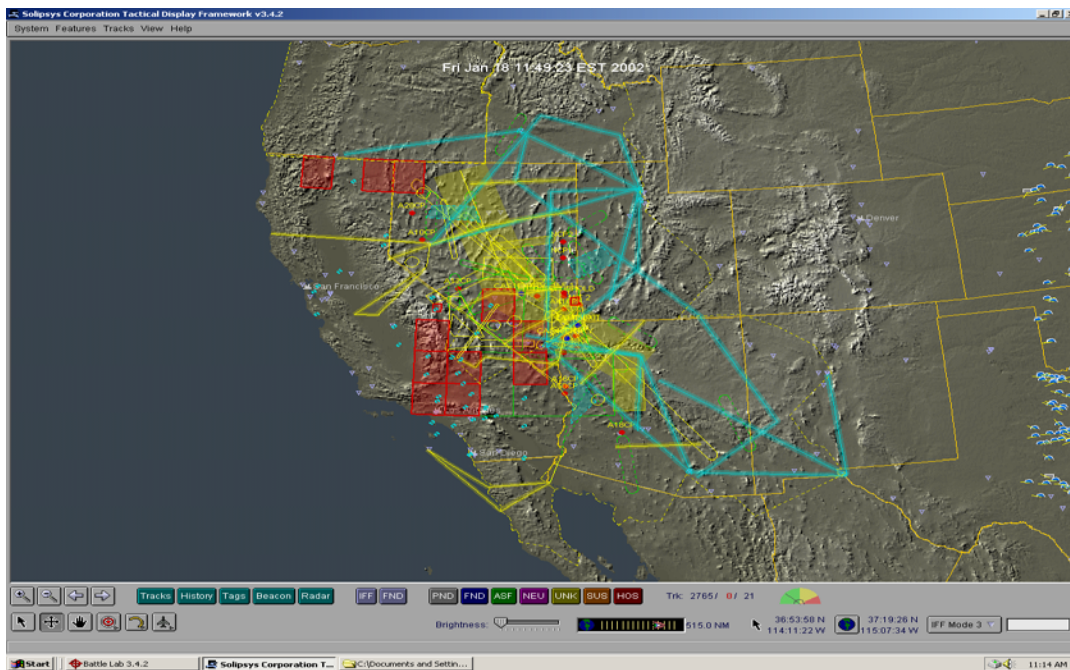


Figure 3 - Automatic ACO/ATO Overlay Creation in TDF

5.1 The TDF Design Approach

Solipsys engineers are constantly surveying the software industry and evaluating and incorporating promising commercial-off-the-shelf (COTS) products or technologies into its products. To minimize the introduction of ephemeral or unproven technologies, all COTS products are formally evaluated by a team of highly qualified software and system engineers prior to recommendation for incorporation into the TDF. The evaluation criteria includes product manufacturer, industry acceptance, maturity, level and quality of documentation, ease of use, proven performance, and availability of related software tools.

Changes to the API are carefully considered and announced to third-party developers at the code level by the built-in Java method deprecation facility prior to implementation in new releases. Deprecated methods are still supported in subsequent releases but use of them results in compile-time warnings that describe the new protocol or object. Since older API methods are deprecated rather than removed, customers are not forced to upgrade their unique applications until it is convenient for them.

Adherence to and compatibility with widely accepted COTS standards are also a hallmark of the TDF. A concrete example of this design philosophy is the recent incorporation of eXtensible Markup Language (XML) technology into the TDF. XML is quickly becoming the preferred method for exchanging and storing text-based, source-decoupled information. To take advantage of this emerging standard, the TDF development team began incorporating XML into newer versions of the framework for pertinent tasks including persistence, display configuration, data storage and transfer, and object distribution. Transitioning the TDF from proprietary formats to XML has led to greater end-user productivity and created new opportunities for interfacing with other COTS products. Newer versions of the PAD have made extensive use of this new XML capability to configure the TDF for AWACS 40/45 purposes.

Solipsys engineers develop the TDF to use the best commercially available software tools and development practices. A philosophy of integrating well-developed commercial products whenever possible and only writing custom software for those applications that have no commercial equivalent has resulted in huge gains in productivity and reliability, and minimized the cost and time associated with adding new features. This design approach also results in an application that is highly portable and compatible with the majority of current and future software systems.

5.2 The TDF Software Organization

The TDF source code is organized according to specific high-level standards set forth by commercial industry for Java applications. These standards ensure compatibility with any other software that may be locally installed on a user's computer or available through the Internet. All source code is developed using strict guidelines and conventions that maximize reuse and readability. Additionally, programmer-level Hyper-Text Markup Language (HTML) documentation is generated for the entire API using the built-in Javadoc utility and distributed with each new version of the toolkit. Internally, completeness of the API documentation is enforced using automated tools that run nightly, guaranteeing a robust and complete documentation package.

TDF's open component architecture permits the seamless integration of other inexpensive commercial tools designed for the PC market. The TDF application can co-exist and interface with commercial office automation and productivity products as well as video teleconferencing and "net meeting" products to facilitate cooperative planning. Industrial-strength data analysis products such as Data Description's DataDesk® and Wavemetric's Igor® have been easily integrated into the TDF using standard interfaces and file formats. Both of these products support the rapid creation of performance and presentation materials and other data products that can be easily imported into other standard software packages such as Microsoft Word® or Excel®. This quick-look capability facilitates real-time performance assessment for training and exercise hot wash-ups.

The TDF is organized into libraries separated into five major categories: Java/Javax, the TDF Core, Geographic, Tactical, and Project-specific classes. Sun Microsystems' Java/Javax classes provide the foundation on which Solipsys builds its own core libraries. The TDF Core libraries define the framework of the application and lay out the basic design approach into generic objects such as views, models, listeners, and factories. The geographic libraries contain coordinate systems, transforms, projections, and database manipulation methods commonly used to present 2D and 3D geographic data. The tactical libraries provide C2-specific capabilities such as track databases, sensors, histories, communications links, and symbol sets. Project-specific libraries layer additional functionality on top of the TDF and customize it for certain specific applications such as the PAD.

Most objects within the TDF are plug-ins loaded at run-time using the built-in Dynamic Loader. The Dynamic Loader operates both at system startup and during execution to load and unload various types of plug-in objects. A plug-in object may be an entire application or a single graphical entity such as a slaved overlay. A path or collection of paths can be passed into the application upon startup to specify where the Dynamic Loader should look for plug-ins. This plug-in concept permits dynamic reconfiguration of the TDF at execution time. An execution-time discovery mechanism is also available to execute and verify Java classes on the fly. The Dynamic Loader and the Java Beans paradigm facilitate the creation of powerful editors for arbitrary data with a minimum of development time.

5.2.1 Concept of Execution

From a high-level, the TDF receives messages and tracks data from external sources. It also acquires geographic and imagery data from binary files. The TDF then displays these varying forms of information to the operator in geographical and tabular formats. The typical method of communication is by standard network connections (Transmission Control Protocol/Internet Protocol [TCP/IP], User Datagram Protocol [UDP]); however, other forms of communication, such as Common Object Request Broker Architecture (CORBA), are also supported. The TDF makes the processing logic parallel whenever possible via the use of multiple threads of execution. The multi-threaded nature of the TDF allows the user display to remain interactive while significant processing takes place in the background.

5.2.2 Model/View

There are three major design paradigms that have been used in the design of the TDF: Model/View, Object/Listener, and Factory. The model/view paradigm strives to separate the underlying data, known as the model, from the presentation of the data, known as the view. Although simple in concept, this powerful design technique leads to modular, maintainable code that permits rapid yet isolated addition of new features. The model provides the data representation of the object and is the basic building block of the design. The view provides a visual representation of the model in a given format such as an editor, text readout, or 2D and 3D display. Because the model is completely independent of the view, new views can be introduced without affecting the model, other views, or system performance. **Figure 4** graphically depicts the separation of the model and examples of several distinct views.

5.2.3 Object/Listener

The object/listener paradigm, depicted in **Figure 5**, decouples the objects interested in certain events (button clicks, mouse motion, state changed, etc.) from the object generating the events. A common analogy is a public speaker and an audience of listeners. The speaker is completely decoupled from the audience and delivers information, known as events in object-oriented design, to audiences of varying sizes. Each member of the audience can choose to listen to none, some, or all of the information that is being presented by the speaker. The speaker's performance is unaffected by the number of listeners and each listener is completely independent of all other listeners. Listeners can individually decide to react to a given piece of information or ignore it altogether. When used as a software design pattern, the object/listener paradigm maximizes object decoupling and leads to a clean design that is extensible and readily modifiable.

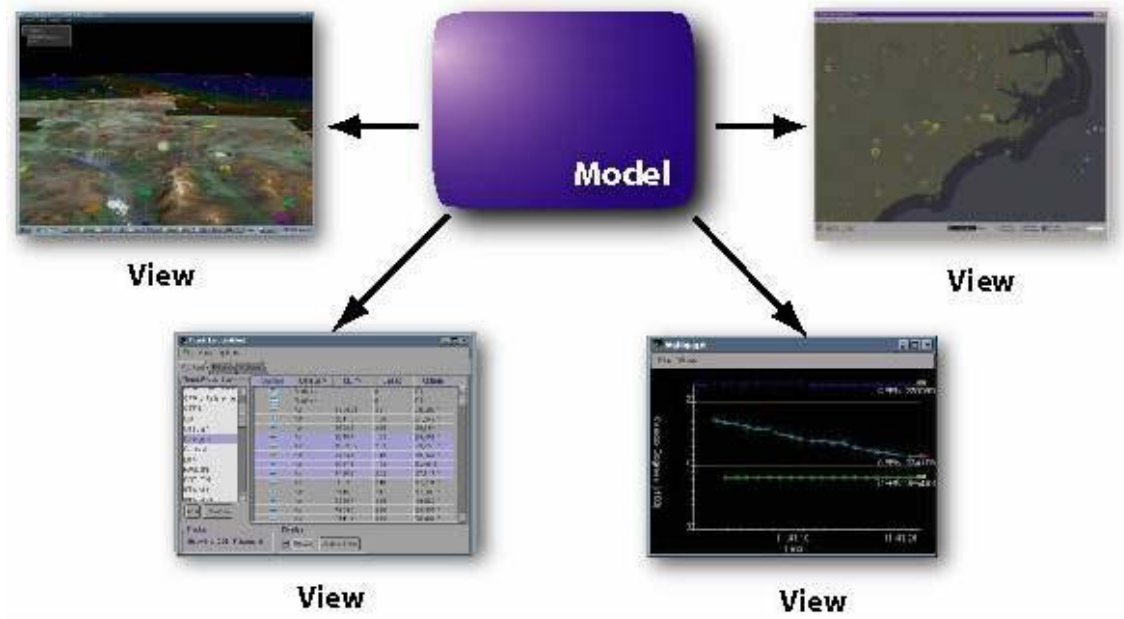


Figure 4 - TDF Model/View Design Pattern

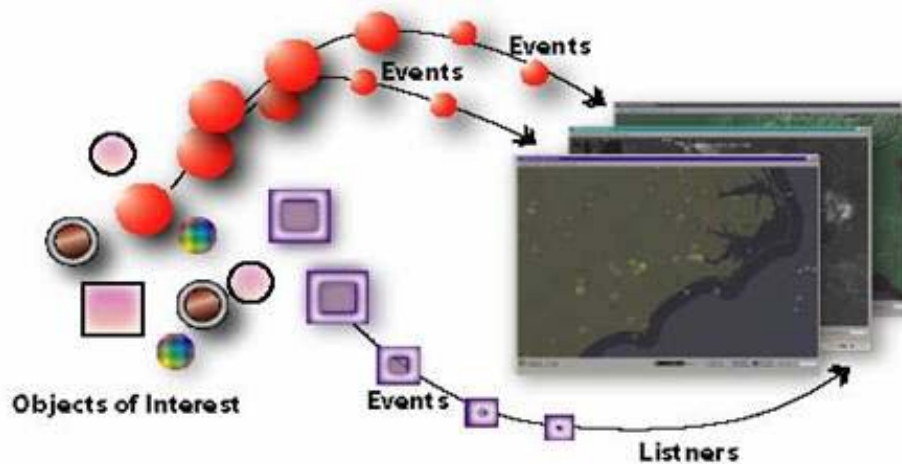


Figure 5 - TDF Object/Listener Design Pattern

The model/view and object/listener paradigms are complementary and often interact with each other within the TDF. For instance, the tactical situation main display simultaneously acts as a view and a listener in that it visually represents an underlying series of data models and updates itself based on events generated by the objects it is displaying.

5.2.4 Factory Paradigm

To efficiently produce commonly used objects such as tracks and sensor plots, the TDF uses the factory paradigm. Objects are run-time representations of data and the functions or methods used to manipulate the data. Since objects are created and destroyed constantly during the course of an application's lifetime, it is important to have an efficient and well-tested mechanism to produce objects on a just-in-time basis. An object factory is a software construct that permits specialized objects to be created quickly and correctly and makes them available to threads of execution upon request. **Figure 6** visually depicts the abstract nature of a software factory.

The three basic design patterns just described overlap frequently and contribute immensely to the portability, flexibility, efficiency, and effectiveness of the TDF.

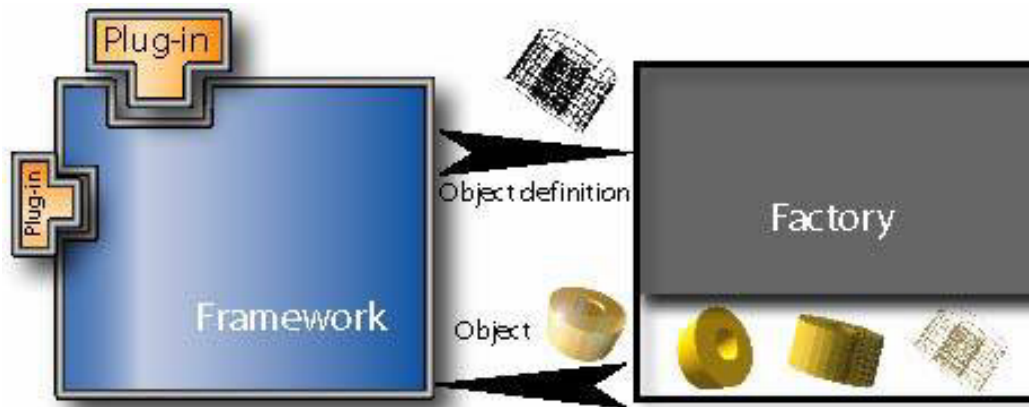


Figure 6 - TDF Factory Design Paradigm

5.2.5 User Features

Since TDF has been under development since 1997, it is a mature product that offers a wide array of user features as shown in **Figure 7** below. The hallmark of TDF is its flexibility on both a software development and user level. Users can dynamically modify nearly every textual and graphical display to suite their needs and save profiles for instant recall of previous settings. Software developers has access to every data element through a comprehensive object-oriented API that provides backward compatibility with prior versions.

6 Conclusion

The USAF has based its future C2 development on the Common Battle Management Software (CBMS) acquisition strategy. Key to the success of CBMS is the development and fielding of an intuitive common look-and-feel. TDF has been embraced by many within the USAF as a key software enabler for achieving the CBMS architectural vision. This Phase I SBIR contract has allowed USAF programs and communities previously unexposed to TDF to learn about the underlying technology.

The success of TDF in the marketplace has shown that Java-based plug-in technology is easily adapted to a variety of applications. Our target markets for the future include:

- Domestic and foreign sales to all military services to include National Military and Civilian Command Centers
- Homeland Security/Defense Modernization
- Federal Aviation Agency

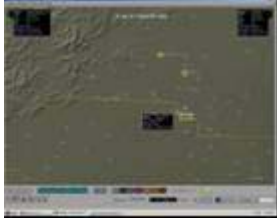
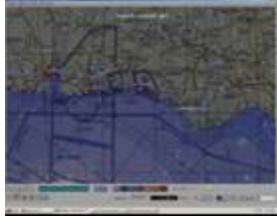

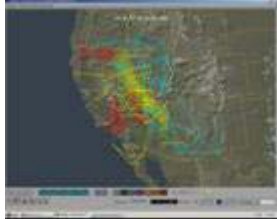
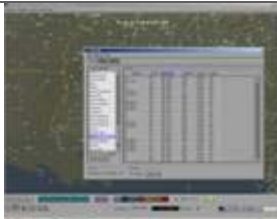

<ul style="list-style-type: none"> • Terrain and imagery data underlays sensor track and plot data as required • Tag data content/format are user-configurable • Altitude tags and track histories data can track friendly resources and potential hostile entities 		<ul style="list-style-type: none"> • Charts can be displayed and turned on and off automatically by range scale selection • Chart data can be augmented by user-drawn overlays • Roads and infrastructure can also be shown with real-time data overlaid 	
<ul style="list-style-type: none"> • An Inset Magnifier helps clarify complex crisis operations • Magnifier can operate on a separate two-headed display in COTS PC, if equipped 		<ul style="list-style-type: none"> • Complex overlays can be built, managed, or imported • Ancillary databases such as Digital Aeronautical Flight Information File (DAFIF) airways can be overlaid on maps • Local nation data and map products can be displayed within the TDF 	
<ul style="list-style-type: none"> • Track/resource lists can be customized and maintained for different missions • Track/resource list can be on main display or auxiliary display using COTS two-port graphics card 		<ul style="list-style-type: none"> • Automatic alert features can identify critical air or ground space violations within complex pictures • A wide range of filter options allow role-based and task-based displays 	

Figure 7 - High-Level Features of TDF

- Disaster and Crisis Management for Local Governments and Emergency Services
- U.S. Customs Services and Border Patrol
- Counter-Drugs Operations
- Time-Critical Targeting (TCT)

Solipsys intends to continue to improve the TDF software and incorporate new features based on user input and market trends. We are hopeful that TDF will become a cornerstone in the USAF's CBMS acquisition strategy.